Beyond Validation: using programmed diagnostics to learn about, monitor, and successfully complete your DH project

Schema-based validation of XML documents has long been a fundamental tool for quality control in digital edition projects, and the emergence of richer schema languages and adjuncts such as Schematron has greatly improved the constraints and controls available to XML authors and encoders (Jacinto et al. 2002). However, schema-based validation typically takes place at the document level, whereas “most programs that use XML require information that is not encoded in the XML instance or in the schema that governs it” (Vorthmann & Robie 2001). The modern digital edition project typically consists of multiple documents with large numbers of pointers between them: links between named entities and personographies, placeographies and bibliographies; pointers to external documents and fragments, images and other media; and similar complex interrelationships within the collection, and to external resources and authorities. These relationships need to be tested, checked, and validated too, but it is impractical to do this using document-level schemas. As Durand et al. (2009) point out, “such testing requirements are in fact closer to conventional system or software testing requirements than to document testing in a narrow sense.” Most large- and medium-scale projects develop their own methods, programmed and/or impromptu, for addressing these problems, and these have been quite well-described and documented for enterprise-level and corporate contexts,[1] but little has been published on project-level diagnostic testing for XML-based digital edition collections.[2]

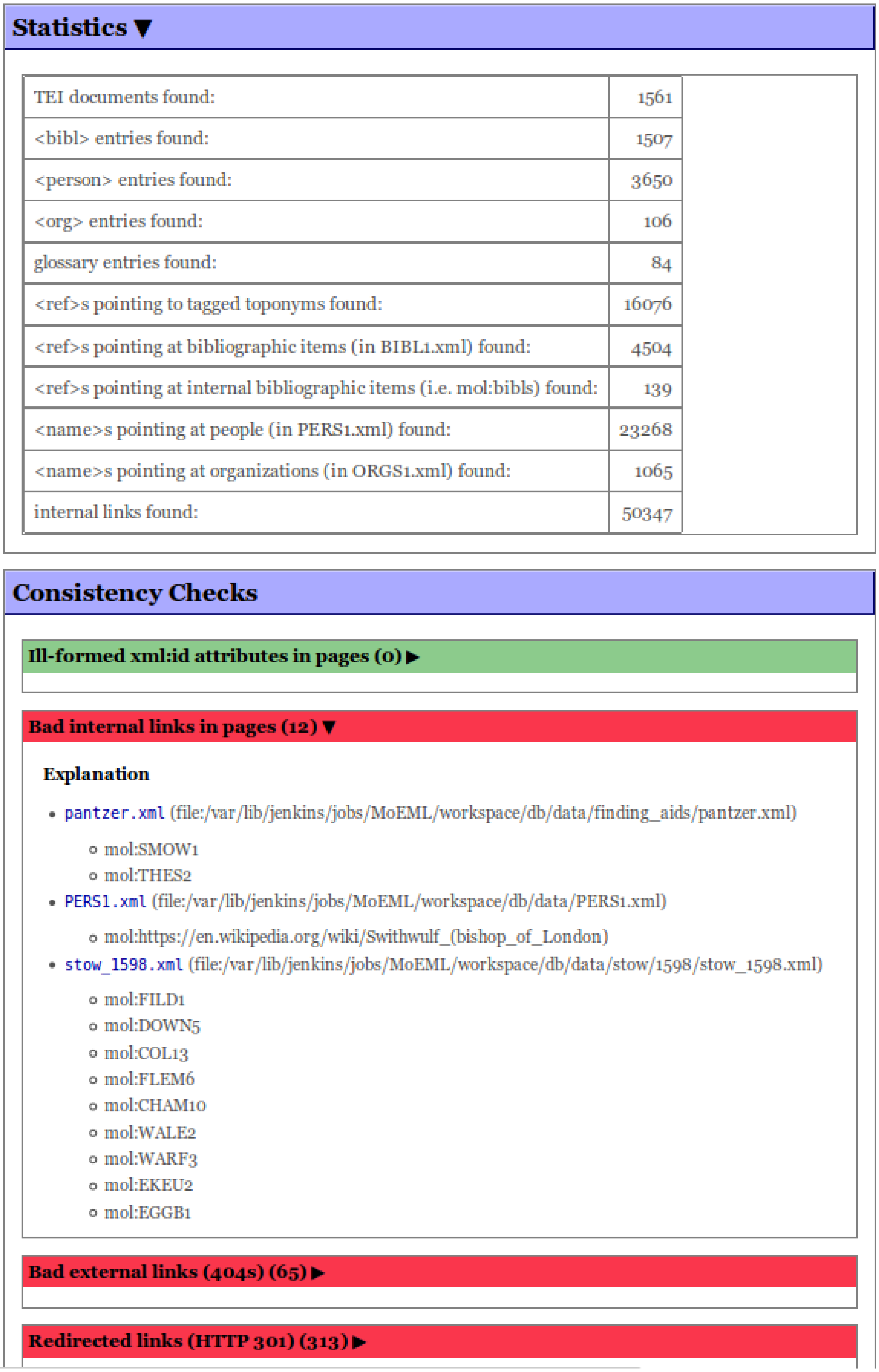
In our work as part of Endings, an umbrella project that comprises four diverse digital edition projects from different fields, we have been developing a structured approach to implementing methods for checking and enforcing project correctness, consistency, and coherence, which we will describe in this paper. Influenced no doubt by Star Trek, we have long referred to these processes as “diagnostics”, and in our description we follow the franchise tradition detailed in Sternbach and Okuda (1991) in dividing diagnostics into levels; however, we depart from convention in ordering our levels from most granular/least comprehensive up to the most general. For each level, we provide real examples of processes run on one of our projects.

We stress that these diagnostics are built on top of a solid basis of RelaxNG and Schematron schemas. In the case of our projects, we use highly-customized versions of the TEI schema (all TEI-compliant) in addition to project-specific Schematron rules, which not only police tagging practices (e.g. enforcing the use of private URI schemes in pointing attributes, and checking the presence of appropriate custom dating attributes for pre-Gregorian dates), but also style guide

[1] For instance, see the papers (particularly Waldt 2012) presented at the International Symposium on Quality Assurance and Quality Control in XML, http://www.balisage.net/Proceedings/vol9/contents.html.

[2] Rahtz (2007) hints at project-level consistency checking, suggesting that editors do not “rely on the schema” and find alternatives “beyond Schema and DTDs.”

rules such as prohibiting the use of straight apostrophes in document text nodes (excepting computer code samples). Our diagnostic processes normally take the form of ant scripts and XSLT transformations, and are run on a Jenkins Continuous Integration server; every time changes are committed to a project repository, the Jenkins server checks out the changes, validates all documents, and runs the entire set of diagnostics processes, providing the results in the form of a public web page such as this one:



Statistics ▼

TEI documents found:	1561
<bibl> entries found:	1507
<person> entries found:	3650
<org> entries found:	106
glossary entries found:	84
<ref>s pointing to tagged toponyms found:	16076
<ref>s pointing at bibliographic items (in BIBL1.xml) found:	4504
<ref>s pointing at internal bibliographic items (i.e. mol:bibls) found:	139
<name>s pointing at people (in PERS1.xml) found:	23268
<name>s pointing at organizations (in ORGS1.xml) found:	1065
internal links found:	50347

Consistency Checks

Ill-formed xml:id attributes in pages (0) ▶

Bad internal links in pages (12) ▼

Explanation

- `pantzer.xml` (file:/var/lib/jenkins/jobs/MoEML/workspace/db/data/finding_aids/pantzer.xml)
    - mol:SMOW1
    - mol:THES2
- `PERS1.xml` (file:/var/lib/jenkins/jobs/MoEML/workspace/db/data/PERS1.xml)
    - mol:https://en.wikipedia.org/wiki/Swithwulf_(bishop_of_London)
- `stow_1598.xml` (file:/var/lib/jenkins/jobs/MoEML/workspace/db/data/stow/1598/stow_1598.xml)
    - mol:FILD1
    - mol:DOWN5
    - mol:COL13
    - mol:FLEM6
    - mol:CHAM10
    - mol:WALE2
    - mol:WARF3
    - mol:EKEU2
    - mol:EGGB1

Bad external links (404s) (65) ▶

Redirected links (HTTP 301) (313) ▶

Figure 1: A diagnostics output page from the *Map of Early Modern London* project.

In combination with this paper, which is intended to be a useful primer and guide, we have developed a Diagnostics project hosted on GitHub (http://github.com/projectEndings/diagnostics) that can be used by researchers whose digital edition projects have grown to the point where ad hoc manual checking has become impractical. This tool provides generic referential integrity checking that can be applied to any set of TEI XML files.

Level 1

Level 1 diagnostics provide project-level, as opposed to document-level, consistency checking to establish the internal coherence of the project, primarily through ensuring referential integrity.[3] This includes checking for non-existent pointers, duplicate @xml:ids across the project, and erroneously encoded references (e.g. tagging a place name as a bibliography reference). Ensuring referential integrity is particularly complex for projects that use “abbreviated pointers” to facilitate internal linking,[4] since it may not be obvious to the encoder which resource is being referenced by a pointer. Thus, the first level of diagnostics checks both whether or not an object pointed to actually exists *and* whether or not the markup correctly represents the relationship between the element and the target resource. For instance, to check all instances of the relationship shown in Fig. 2, a number of different tests are actually done:

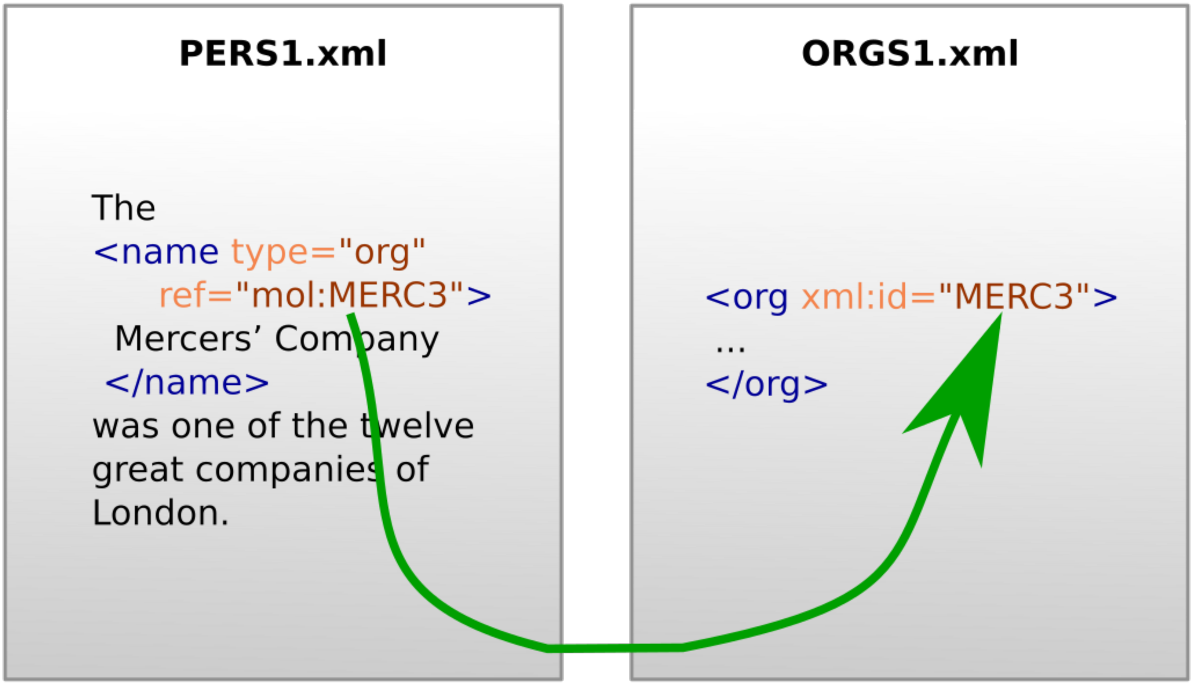


Figure 2: a simple referential integrity check.

[3] We borrow the phrase “referential integrity” from the MLA’s “Guiding Questions for Vetters of Scholarly Editions” (2011), which advises peer-reviewers of digital editions that link to multiple databases to see if “referential integrity [is] enforced within the database(s).”
[4] See TEI Consortium (2016), http://www.tei-c.org/release/doc/tei-p5-doc/en/html/SA.html#SAPU.

1. Every <name type=”org”> points at an @xml:id which exists in the project.
2. The element pointed at by <name type=”org”> is an <org> element in the ORGS1.xml document.
3. Every <name> element which points at an <org> element in ORGS1.xml has @type=”org”.

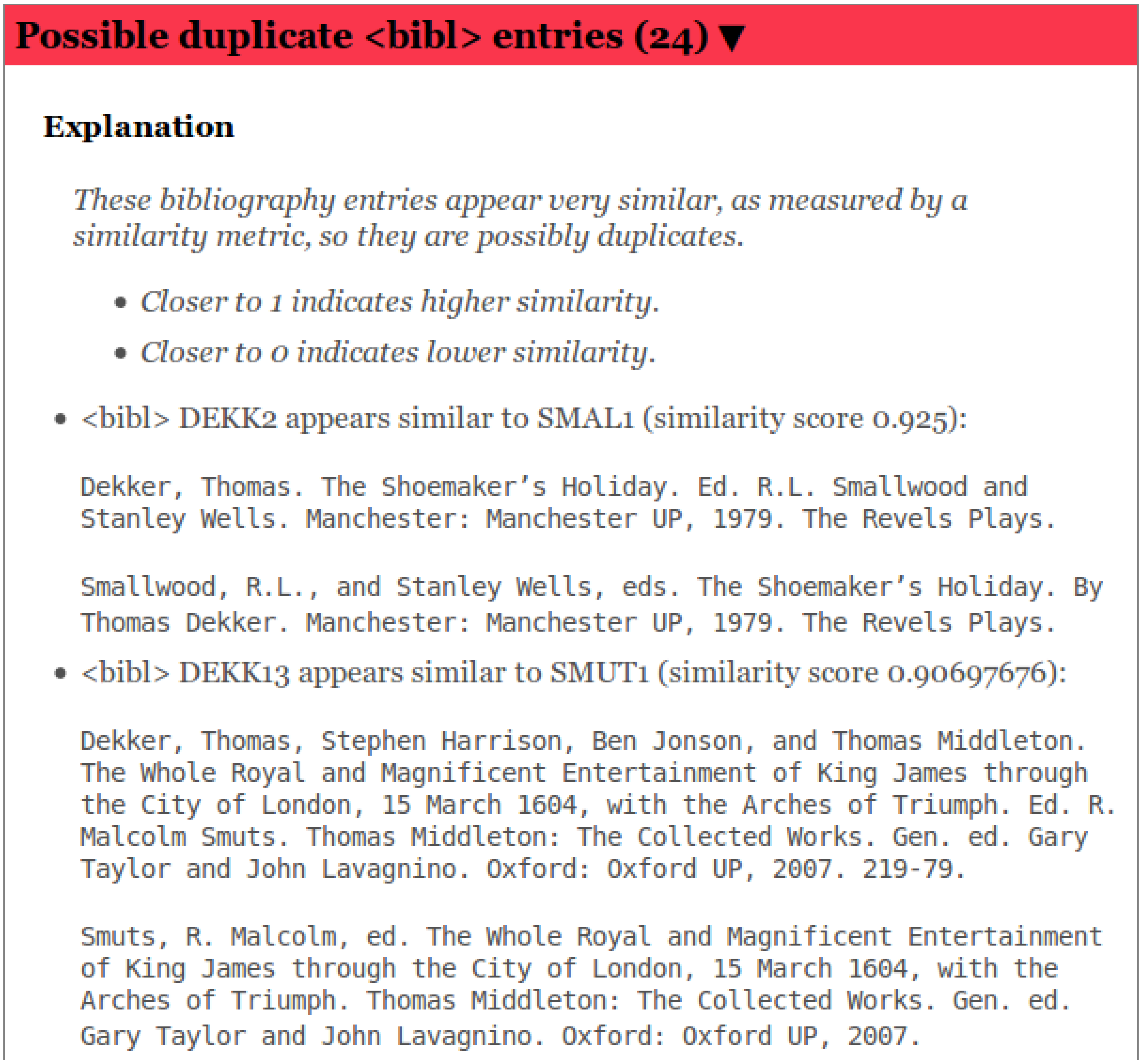
For small-scale projects, this kind of referential integrity check could be accomplished with Schematron, since a Schematron rule using XPath 2.0 can read external documents, but for a project of any significant size, this is impractical. For example, Schematron checks to confirm the rules above may add around six seconds to document validation in the Oxygen XML Editor, causing frustration for editors, while simply checking that a linked location exists would require the processing of over a thousand files in this project, since each location is a distinct file.

Level 2

While Level 1 diagnostics generally focus on coherence and consistency, Level 2 is more concerned with completeness. Level 2 diagnostics provide progress analysis, generate to-do lists, and identify situations that may indicate error, but require human judgement. These include cases in which:

- Two bibliography or personography entries appear sufficiently similar that they may be duplicates.
- Several <name> elements point to the same authority record, but the text of one of them is significantly different from the others, so it may point at the wrong target.
- A document in the project is not linked from anywhere else, and therefore cannot be “reached”.

Such issues cannot be automatically rectified—they are not necessarily errors—but they must be examined. Figure 3 shows an example of the first check, which uses a similarity metric to identify potential duplicate bibliography entries.



Possible duplicate <bibl> entries (24) ▼

Explanation

These bibliography entries appear very similar, as measured by a similarity metric, so they are possibly duplicates.

- *Closer to 1 indicates higher similarity.*
- *Closer to 0 indicates lower similarity.*

- <bibl> DEKK2 appears similar to SMAL1 (similarity score 0.925):

```
Dekker, Thomas. The Shoemaker’s Holiday. Ed. R.L. Smallwood and
Stanley Wells. Manchester: Manchester UP, 1979. The Revels Plays.

Smallwood, R.L., and Stanley Wells, eds. The Shoemaker’s Holiday. By
Thomas Dekker. Manchester: Manchester UP, 1979. The Revels Plays.
```

- <bibl> DEKK13 appears similar to SMUT1 (similarity score 0.90697676):

```
Dekker, Thomas, Stephen Harrison, Ben Jonson, and Thomas Middleton.
The Whole Royal and Magnificent Entertainment of King James through
the City of London, 15 March 1604, with the Arches of Triumph. Ed. R.
Malcolm Smuts. Thomas Middleton: The Collected Works. Gen. ed. Gary
Taylor and John Lavagnino. Oxford: Oxford UP, 2007. 219-79.

Smuts, R. Malcolm, ed. The Whole Royal and Magnificent Entertainment
of King James through the City of London, 15 March 1604, with the
Arches of Triumph. Thomas Middleton: The Collected Works. Gen. ed.
Gary Taylor and John Lavagnino. Oxford: Oxford UP, 2007.
```

Figure 3: Results of a Level 2 diagnostic check that attempts to identify duplicate bibliography entries.

At Level 2, we also generate to-do lists for specific sub-projects, providing a set of tasks for the project team to focus on in order to reach a milestone or publish a particular document. The definition of “done” for a specific document may transcend the document itself. For instance, before we deem a particular edition of a text publishable, we may require that all authority records (people, places, publications) linked from that document are themselves complete, so the to-do list for a given document may require work in a variety of other documents in the project.

Level 3

Armed with a comprehensive set of Level 1 and Level 2 diagnostics, and assuming our data is managed using a version-control repository such as Subversion or Git, we can now generate diachronic views of the project's progress. A script can check out a sequence of incarnations of the project, weekly over a period of months, for instance, and run the entire current diagnostic suite against it; we can then combine these snapshots to get a clear sense of how our work is proceeding. This also means that every time we develop a new diagnostic procedure, we can apply it to the entire history of the project to see the trajectory of project work with respect to the datapoint in question. Two examples, this time from the Nxaʔamxcín Dictionary project,[5] appear in Figs 4 and 5 below. Figure 4 shows the number of completed dictionary entries in orange, rising steadily over a period of 18 months, and the number of occurrences of a known problem: duplicate instances of the same gloss. These duplicates rise along with the number of entries until October 2016, when this issue was added to our diagnostics process, and the encoders were able to address it.

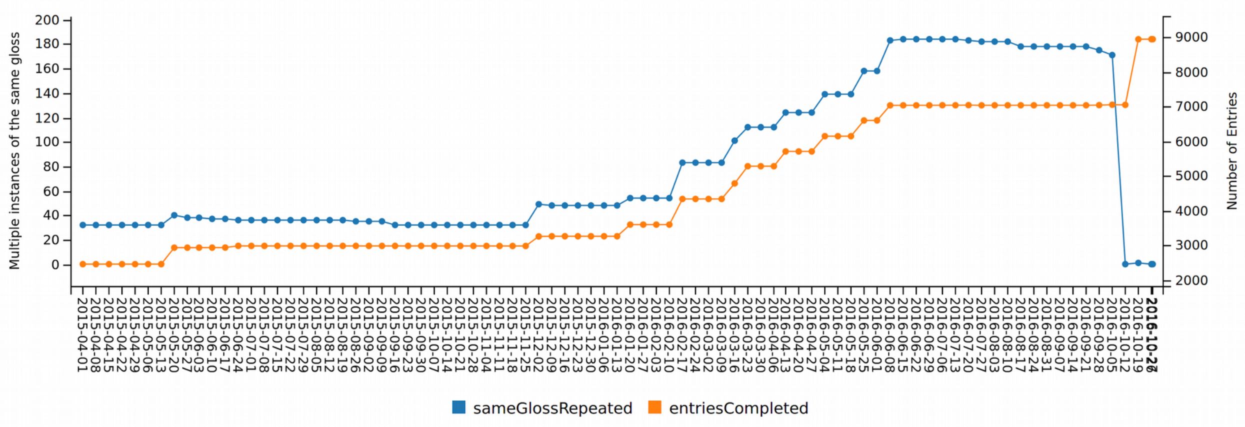


Figure 4: The number of instances of duplicate glosses, tracked against completed entries, in the Nxaʔamxcín Dictionary project.

Fig. 5 shows cases of broken cross-references, which also tend to increase along with the number of completed entries, but we can see from the graph that the issue was aggressively addressed in two separate campaigns in fall 2015 and summer 2016. New instances continue to appear, however.

[5] The Nxaʔamxcín Dictionary is an indigenous dictionary project described in detail in Czaykowska-Higgins, Holmes, and Kell (2014).

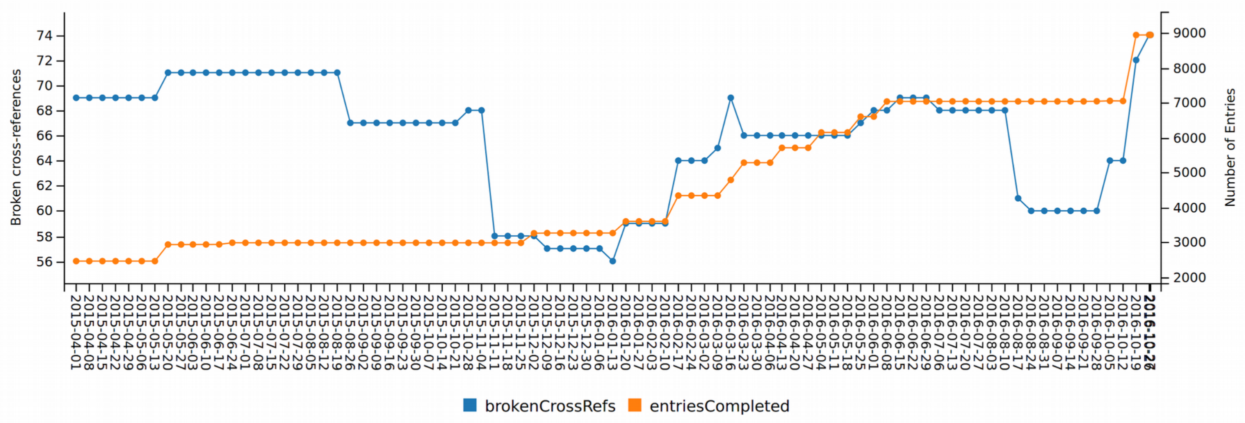
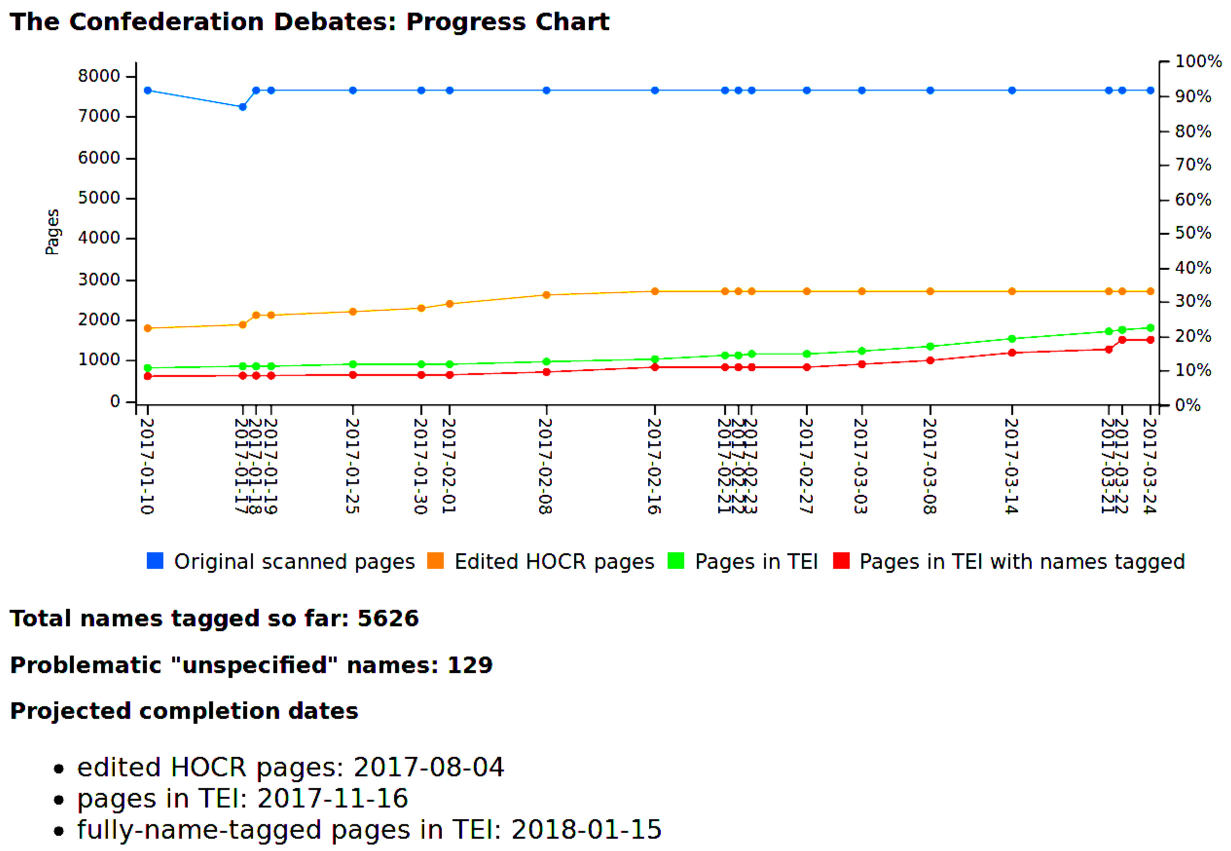


Figure 5: The number of broken cross-references, tracked against completed entries.

Fig. 6, from a different project, shows how this approach can be used to forecast completion dates for tasks in a project based on the progress rate so far.



The Confederation Debates: Progress Chart

Total names tagged so far: 5626

Problematic "unspecified" names: 129

Projected completion dates

- edited HOCR pages: 2017-08-04
- pages in TEI: 2017-11-16
- fully-name-tagged pages in TEI: 2018-01-15

Figure 6: Diachronic diagnostics used to project task completion dates.

Conclusion

As Matthew Kirschenbaum (2009) tells us, there “is no more satisfying sequence of characters” than “Done.” The overall purpose of a digital edition project is to finish and publish the edition, and this requires not only that the document-level encoding be valid, but also that the entire dataset be coherent, consistent, and complete. Programmed diagnostics enable projects to enforce coherence and consistency, manage the workflow effectively, and measure their progress towards completeness.

References

Czaykowska-Higgins, Ewa, Martin Holmes, and Sarah Kell. 2014. “Using TEI for an Endangered Language Lexical Resource: The Nxaʔamxcín Database-Dictionary Project.” *Language Documentation & Conservation* 8: 1–37.

“Guidelines for Editors of Scholarly Editions.” 2016. *Modern Language Association*. Accessed September 15. https://www.mla.org/Resources/Research/Surveys-Reports-and-Other-Documents/Publishing-and-Scholarship/Reports-from-the-MLA-Committee-on-Scholarly-Editions/Guidelines-for-Editors-of-Scholarly-Editions.

“Guiding Questions for Vetters of Scholarly Editions.” 2011. *Modern Language Association*. Accessed October 21. https://www.mla.org/content/download/3201/81158/cse_guidelines_2011.pdf.

Jacinto, Marta Henriques, Giovani Rubert Librelotto, José Carlos Ramalho, and Pedro Rangel Henriques. 2002. “Constraint specification languages : comparing XCSL, Schematron and XML-Schemas.” http://repositorium.sdum.uminho.pt/handle/1822/619.

Kirschenbaum, Matthew. 2009. “Done: Finishing Projects in the Digital Humanities.” *DHQ* 3 (2). http://digitalhumanities.org:8081/dhq/vol/3/2/000037/000037.html.

“Proceedings of the International Symposium on Quality Assurance and Quality Control in XML.” 2012. http://www.balisage.net/Proceedings/vol9/contents.html.

Rahtz, Sebastian. 2007. “Technology Overview and Discussion: Data Capture, Editing, and Schemas.” Oxford, February 13. http://tei.it.ox.ac.uk/Talks/2007-02-13-oucs/talk-editing.xml.

Sternbach, Rick, and Michael Okuda. 1991. *Star Trek, the next Generation: Technical Manual*. New York: Pocket Books. http://catalog.hathitrust.org/api/volumes/oclc/24648561.html.

Vorthmann, Scott, and Jonathan Robie. 2001. “Beyond Schemas: Schema Adjuncts and the Outside World.” *Markup Languages: Theory & Practice* 2 (3): 281–94.

Waldt, Dale. 2012. “Quality Assurance in the XML World: Beyond Validation.” Accessed September 15. http://www.balisage.net/Proceedings/vol9/author-pkg/Waldt01/BalisageVol9-Waldt01.html.